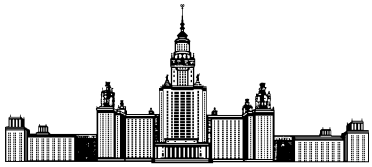
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

КУРСОВАЯ РАБОТА СТУДЕНТА 417 ГРУППЫ

«Комбинаторные оценки обобщающей способности и методы их вычисления»

Выполнил:

студент 4 курса 417 группы

Соколов Евгений Андреевич

Научный руководитель:

д.ф-м.н., доцент

Воронцов Константин Вячеславович

Заведующий кафедрой

Математических Методов

Прогнозирования, академик РАН _____ Ю. И. Журавлёв

К защите допускаю

«_____» _____ 2012 г.

К защите рекомендую

«_____» _____ 2012 г.

Москва, 2012

Содержание

1 Введение

1.1 Основные определения

Пусть задано конечное множество $\mathbb{X} = \{x_1, \ldots, x_L\}$, называемое *генеральной выборкой*, и целевая функция $y : \mathbb{X} \to \mathbb{Y}$. Пусть также задано множество $\mathcal{A}$, элементы которого называют *алгоритмами*. Алгоритм $a \in \mathcal{A}$ сопоставляет объекту $x \in \mathbb{X}$ некоторое значение из множества $\mathbb{Y}$, $a : \mathbb{X} \to \mathbb{Y}$. Также предполагается, что задана функция $I : \mathcal{A} \times \mathbb{X} \to \{0, 1\}$, называемая *индикатором ошибки*. $I(a, x)$ принимает значение 1, если алгоритм a допускает ошибку на объекте x, и значение 0 в противном случае.

Вектором ошибок алгоритма a называется бинарный вектор $\vec{a} = (I(a, x_i))_{i=1}^{L}$. Везде далее под «алгоритмом» будем понимать не само отображение, а его вектор ошибок. Также будем считать, что $\mathcal{A}$ — это множество бинарных векторов.

Метод обучения — это функция, строящая по подмножеству полной выборки алгоритм из заданного семейства: $\mu : 2^{\mathbb{X}} \to \mathcal{A}$.

Будем рассматривать задачу обучения по прецедентам в следующей постановке. Пусть на этапе обучения известна *обучающая выборка* $X \subset \mathbb{X}$ длины ℓ. По обучающей выборке с помощью заданного метода обучения μ выбирается алгоритм μX из семейства $\mathbb{A}$. После того, как обучение закончено, становится известной *скрытая выборка* $\bar{X} = \mathbb{X} \setminus X$, и к ней применяется выбранный алгоритм μX. Длину контрольной выборки будем обозначать через $k = L - \ell$.

Числом ошибок алгоритма a на выборке $X \subset \mathbb{X}$ называют величину

$$n(a, X) = \sum_{x \in X} I(a, x)$$

Долей ошибок алгоритма a на выборке $X \subset \mathbb{X}$ называется величина

$$\nu(a, X) = \frac{n(a, X)}{|X|}$$

Уклонением частот ошибок алгоритма a на двух выборках X и $\bar{X} = \mathbb{X} \setminus X$ называется величина

$$\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X)$$

Пусть задан некоторый вещественный параметр $\varepsilon \in [0, 1)$, называемый *порогом переобучения*. Говорят, что алгоритм a переобучается на разбиении $(X, \bar{X})$, если

$$\delta(a, X) \geq \varepsilon$$

Аналогично, метод μ переобучается на разбиении $(X, \bar{X})$, если

$$\delta(\mu X, X) \geq \varepsilon$$

Определение 1.1. *Вероятностью переобучения метода μ называется величина*

$$Q_\varepsilon(\mu, \mathbb{X}) \equiv \mathsf{P}[\delta(\mu X, X) \geq \varepsilon] = \frac{1}{C_L^\ell} \sum_{(X,\bar{X})} [\delta(\mu X, X) \geq \varepsilon]$$

Определим некоторые комбинаторные величины, которые понадобятся нам в дальнейшем.

Гипергеометрическая функция вероятности:

$$h_L^{\ell,m}(s) = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$$

Гипергеометрическая функция распределения:

$$H_L^{\ell,m}(s) = \sum_{i=0}^{\min(s,\ell,m)} h_L^{\ell,m}(i)$$

Подробное описание этих величин можно найти в ([?]).

В данной работе большое внимание будет уделено семейству *линейных классификаторов*. Пусть объекты выборки представляют собой точки в некотором евклидовом пространстве: $\mathbb{X} \subset \mathbb{R}^d$. Тогда семейство линейных классификаторов $\mathcal{A}_h$ — это множество всех гиперплоскостей, разделяющих данную выборку

$$\mathcal{A}_h = \{a_w(x) = \text{sign}(\langle w, x\rangle + w_0) \mid w \in \mathbb{R}^d, w_0 \in \mathbb{R}\}$$

Как было сказано выше, множество алгоритмов мы будем отождествлять с множеством векторов ошибок этих алгоритмов. Это означает, что множество $\mathcal{A}_h$ мы будем отождествлять с множеством всех бинарных векторов, соответствующих разделению выборки на две части гиперплоскостью.

1.2 Представление семейства алгоритмов графом

Введем на множестве алгоритмов отношение частичного порядка $\prec$:

$$a \leq b \;\Leftrightarrow\; (\forall x \in \mathbb{X}\; I(a,x) \leq I(b,x))$$

Если $a \leq b$ и при этом $\rho(a,b) = 1$ (здесь ρ — это хэммингово расстояние), то будем говорить, что a *предшествует* b и записывать $a \prec b$.

Определение 1.2. *Графом расслоения-связности семейства алгоритмов* $\mathcal{A}$ *называется ориентированный граф* $G = (V, E)$ *с множеством вершин* $V = \mathcal{A}$ *и множеством ребер* $E = \{(a,b) \mid a \prec b\}$.

Слоем графа расслоения-связности называется множество алгоритмов, допускающих одинаковое число ошибок: $A_m = \{a \in \mathcal{A} \mid n(a,\mathbb{X}) = m\}$. Граф расслоения-связности является многодольным, доли соответствуют слоям A_m, ребрами могут соединяться только соседние слои. В частности, из многодольности графа следует его двудольность.

Если две вершины графа a и b соединены ребром (где $a \prec b$), то векторы ошибок алгоритмов a и b отличаются лишь в одном элементе. Это позволяет поставить в соответствие каждому ребру (a,b) объект $x_{ab} \in \mathbb{X}$ такой, что $I(a,x_{ab}) = 0$ и $I(b,x_{ab}) = 1$.

Верхней окрестностью алгоритма a называется множество $C^+(a) = \{b \in \mathbb{A} \mid (a,b) \in E\}$. Аналогично, *нижней окрестностью* называется множество $C^-(a) = \{b \in \mathbb{A} \mid (b,a) \in E\}$. Элементы верхней и нижней окрестностей алгоритма a будем называть верхними и нижними соседями соответственно.

Вершина графа расслоения-связности называется *истоком*, если у нее нет входящих ребер.

Верхней связностью $q(a)$ алгоритма a называется число вершин в его верхней окрестности:

$$u(a) = |C_+(a)|$$

Неполноценностью (inferiority) $r(a)$ алгоритма a называется число объектов $x \in \mathbb{X}$, на которых a ошибается, при том, что существует алгоритм $b \prec a$, не ошиба-

ющийся на x:

$$q(a) = \#\{x \in \mathbb{X} \mid I(a,x) = 1,\ \exists b \in \mathbb{A} :\ b \prec a, I(b,x) = 0\}$$

Введем также обозначение для числа ошибок алгоритма a:

$$m(a) = n(a, \mathbb{X})$$

1.3 Оценка расслоения-связности

Пусть $\mathbb{X} = \{x_1, \ldots, x_L\}$ — выборка, $\mathbb{A} = \{a_1, \ldots, a_D\}$ — некоторое семейство алгоритмов. Будем считать, что алгоритмы пронумерованы в порядке неубывания числа ошибок на генеральной выборке.

Определение 1.3. *Метод обучения μ называется пессимистичным методом минимизации эмпирического риска (ПМЭР), если он выбирает алгоритм, допускающий наименьшее число ошибок на обучающей выборке; если таких несколько — выбирает из них алгоритм с наибольшим числом ошибок на генеральной выборке; если и таких несколько, то он выбирает из них алгоритм с наибольшим номером.*

Отметим, что так как алгоритмы отсортированы по числу ошибок, то из всех алгоритмов, минимизирующих эмпирический риск, μ будет просто выбирать алгоритм с наибольшим номером.

С использованием введенных ранее характеристик алгоритмов, основанных на графе расслоения-связности, была получена следующая оценка вероятности переобучения для ПМЭР:

Теорема 1 (Воронцов, Решетняк, Ивахненко, 2010)**.** *Для пессиместичного метода минимизации эмпирического риска μ и любых $\mathbb{X}$, $\mathcal{A}$ и $\varepsilon \in (0,1)$*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{i=1}^{D} \frac{C_{L-u-q}^{\ell-u}}{C_L^\ell} \mathcal{H}_{L-u-q}^{\ell-u,\ m-q}\left(\frac{\ell}{L}(m - \varepsilon k)\right)$$

Отметим, что вклад алгоритма a в данную оценку экспоненциально убывает с ростом $u(a)$ и $q(a)$.

2 Улучшенная оценка расслоения-связности

Напомним некоторые определения и предположения, описанные ранее. Пусть $\mathbb{X} = \{x_1, \ldots, x_L\}$ — выборка, $\mathbb{A} = \{a_1, \ldots, a_D\}$ — некоторое семейство алгоритмов. Мы считаем, что алгоритмы пронумерованы в порядке неубывания числа ошибок на генеральной выборке.

Определение 2.1. *Метод обучения μ называется пессимистичным методом минимизации эмпирического риска (ПМЭР), если он выбирает алгоритм, допускающий наименьшее число ошибок на обучающей выборке; если таких несколько — выбирает из них алгоритм с наибольшим числом ошибок на генеральной выборке; если и таких несколько, то он выбирает из них алгоритм с наибольшим номером.*

Из всех алгоритмов, минимизирующих эмпирический риск, μ будет просто выбирать алгоритм с наибольшим номером.

Определим для произвольных двух алгоритмов a_i и a_j множества A_{ij} и B_{ij}:

$$A_{ij} = \{x \in \mathbb{X} \mid I(a_i, x) = 0, I(a_j, x) = 1\}$$
$$B_{ij} = \{x \in \mathbb{X} \mid I(a_i, x) = 1, I(a_j, x) = 0\}$$

Это можно изобразить следующим образом:

$$a_i : (0 \ldots 0\ 0 \ldots 0\ 1 \ldots 1\ 1 \ldots 1)$$
$$a_j : (0 \ldots 0\ \underbrace{1 \ldots 1}_{A_{ij}}\ \underbrace{0 \ldots 0}_{B_{ij}}\ 1 \ldots 1)$$

Лемма 1. *Пусть X - обучающая выборка, μ — пессимистичный метод минимизации эмпирического риска. Тогда:*

$$[\mu X = a_i] = \left(\prod_{j=1}^{i-1} [|X \cap B_{ij}| \leq |X \cap A_{ij}|]\right) \left(\prod_{j=i+1}^{D} [|X \cap B_{ij}| < |X \cap A_{ij}|]\right) \qquad (2.1)$$

Доказательство. Заметим, что если $|X \cap B_{ij}| > |X \cap A_{ij}|$, то a_j допускает на обучающей выборке меньше ошибок, чем a_i. Значит, в этом случае a_i не может быть выбран методом μ. Также, согласно определению метода μ, при равном числе ошибок на выборке он выбирает алгоритм с наибольшим номером.

Значит, для того, чтобы алгоритм a_i был выбран ПМЭР μ на обучающей выборке X, необходимо, чтобы для любого $j = 1, \ldots, D$ было выполнено:

- $|X \cap B_{ij}| \leq |X \cap A_{ij}|$, если $j < i$
- $|X \cap B_{ij}| < |X \cap A_{ij}|$, если $j > i$

Значит, верно следующее неравенство:

$$[\mu X = a_i] \leq \left(\prod_{j=1}^{i-1} [|X \cap B_{ij}| \leq |X \cap A_{ij}|]\right) \left(\prod_{j=i+1}^{D} [|X \cap B_{ij}| < |X \cap A_{ij}|]\right)$$

Покажем, что данное неравенство выполнено и в другую сторону. Действительно, если правая часть равенства (2.1) равна единице, то алгоритм a_i допускает не больше ошибок на X, чем какой-либо другой алгоритм. Более того, не существует алгоритма с номером $j > i$, допускающего столько же ошибок, сколько и a_i. Так как среди всех алгоритмов, минимизирующих эмпирический риск, метод обучения μ выбирает алгоритм с наибольшим номером, то выполнено

$$[\mu X = a_i] \geq \left(\prod_{j=1}^{i-1} [|X \cap B_{ij}| \leq |X \cap A_{ij}|]\right) \left(\prod_{j=i+1}^{D} [|X \cap B_{ij}| < |X \cap A_{ij}|]\right)$$

□

Лемма 2. *Пусть μ — метод пессимистичной минимизации эмпирического риска, a_i и a_s — два произвольных алгоритма из $\mathcal{A}$. Тогда:*

$$\mathsf{P}[\mu X = a_i][\delta(a_i, X) \geq \varepsilon] \leq \sum_{t=0}^{\min(|A_{is}|,|B_{is}|)} \frac{C_{|B_{is}|}^{t} C_{L-u-|B_{is}|}^{l-u-t}}{C_L^l} \mathcal{H}_{L-u-|B_{is}|}^{l-u-t,\ m-|B_{is}|} \left(\frac{l}{L}(m - \varepsilon k) - t\right)$$

Доказательство. С помощью леммы 1 оценим величину $[\mu X = a_i]$ сверху, оставив только те множители, которые соответствуют a_s и верхней полуокрестности:

$$\begin{aligned}
[\mu X = a_i] &\leq ([s \leq i]\,[|B_{is} \cap X| \leq |A_{is} \cap X|] + [s > i]\,[|B_{is} \cap X| < |A_{is} \cap X|]) \times \\
&\quad \times \prod_{j:\ a_j \in C^+(a_i)} [|B_{ij} \cap X| < |A_{ij} \cap X|] \leq \\
&\leq [|B_{is} \cap X| \leq |A_{is} \cap X|] \prod_{j:\ a_j \in C^+(a_i)} [|B_{ij} \cap X| < |A_{ij} \cap X|] \leq \\
&\leq [|B_{is} \cap X| \leq |A_{is}|] \prod_{j:\ a_j \in C^+(a_i)} [|B_{ij} \cap X| < |A_{ij} \cap X|] = \\
&= [|B_{is} \cap X| \leq |A_{is}|] \prod_{j:\ a_j \in C^+(a_i)} [|A_{ij} \cap X| > 0]
\end{aligned}$$

Из последнего неравенства следует, что для того, чтобы был выбран алгоритм a_i, необходимо, чтобы в обучающую выборку попали все объекты из $\bigcup_{a_j \in C^+(a_i)} A_{ij}$, и чтобы из B_{is} в X попало не более $|A_{is}|$ объектов. Исходя из этих соображений, получаем следующую оценку:

$$\mathsf{P}[\mu X = a_i][\delta(a_i, X) \geq \varepsilon] \leq \sum_{t=0}^{\min(|A_{is}|,|B_{is}|)} \frac{C_{|B_{is}|}^{t} C_{L-u-|B_{is}|}^{l-u-t}}{C_L^l} \mathcal{H}_{L-u-|B_{is}|}^{l-u-t,\ m-|B_{is}|} \left(\frac{l}{L}(m - \varepsilon k) - t \right)$$

□

Теорема 2. *Пусть μ — метод пессимистичной минимизации эмпирического риска, S — множество всех истоков графа расслоения-связности. Тогда верна следующая оценка вероятности переобучения:*

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{i=1}^{D} \min_{s \in S} \left\{ \sum_{t=0}^{\min(|A_{is}|,|B_{is}|)} \frac{C_{|B_{is}|}^{t} C_{L-u-|B_{is}|}^{l-u-t}}{C_L^l} \mathcal{H}_{L-u-|B_{is}|}^{l-u-t,\ m-|B_{is}|} \left(\frac{l}{L}(m - \varepsilon k) - t \right) \right\} \tag{2.2}$$

Доказательство. Распишем вероятность переобучения, используя формулу полной вероятности:

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}[\delta(\mu X, X) \geq \varepsilon] = \sum_{i=1}^{D} \mathsf{P}[\mu X = a_i][\delta(a_i, X) \geq \varepsilon]$$

Для различных $a_s \in \mathcal{A}$ можно получить различные оценки для величины $\mathsf{P}[\mu X = a_i][\delta(a_i, X) \geq \varepsilon]$, используя лемму 2. Мы вычислим такие оценки для всех a_s, являющихся истоками графа расслоения-связности, а затем выберем наименьшую из таких оценок:

$$\begin{aligned} Q_\varepsilon(\mu, \mathbb{X}) &= \sum_{i=1}^{D} \mathsf{P}[\mu X = a_i][\delta(a_i, X) \geq \varepsilon] \leq \\ &\leq \sum_{i=1}^{D} \min_{s \in S} \left\{ \sum_{t=0}^{\min(|A_{is}|,|B_{is}|)} \frac{C_{|B_{is}|}^{t} C_{L-u-|B_{is}|}^{l-u-t}}{C_L^l} \mathcal{H}_{L-u-|B_{is}|}^{l-u-t,\ m-|B_{is}|} \left(\frac{l}{L}(m - \varepsilon k) - t \right) \right\} \end{aligned}$$

□

В разделе 3 будет произведено сравнение данной оценки с оценкой (1) и будет показано, что новая оценка точнее на порядок или больше.

2.1 Направления дальнейших исследований

- Обобщение оценки на настоящий граф Хассе;
- Получение оценок, учитывающих связь с двумя истоками или больше.

3 Комбинаторные отступы и отбор объектов

Оценки (1) и (2) зависят от всех алгоритмов семейства $\mathcal{A}$, поэтому для вычисления этих оценок необходимо сначала найти $\mathcal{A}$. Если выборка состоит из большого числа объектов, то построение всего семейства алгоритмов может оказаться крайне трудоемкой задачей. В то же время интерес представляют только алгоритмы из нижних слоев графа расслоения-связности, так как лишь они делают существенный вклад в оценку вероятности переобучения. В данном разделе мы покажем, что можно исключить из выборки некоторые объекты таким образом, что нижние слои графа не изменятся.

Ребро в SC-графе, идущее из вершины a в вершину a', соответствует изменению классификации на одном объекте. Если известно, что объект x не соответствует ни одному ребру в нижних слоях графа, то его можно не рассматривать при обходе. Выясним, что характеризует свойство объекта «иметь ребра в нижних слоях».

Определение 3.1. *Комбинаторным отступом алгоритма a_s на объекте x_0 называется следующая величина:*

$$d(a_s, x_0) = \min\{d \mid \exists a_i : \ I(a_s, x_0) \neq I(a_i, x_0), |B_{is}| = d\}$$

Пусть $S \subset \mathbb{A}$ — множество всех истоков SC-графа семейства $\mathbb{A}$. Определим отступ объекта x как минимальный из отступов всех истоков на нем:

$$d(x) = \min_{a \in S} d(a, x)$$

Будем говорить, что x_0 — *порождающий* объект для алгоритма a, если существует такой алгоритм a', что векторы ошибок a и a' отличаются только на x_0. Очевидно, что объект x_0 является порождающим для a тогда и только тогда, когда из a выходит ребро, соответствующее объекту x_0.

Допустим, мы исключаем из рассмотрения объект x_i. Это соответствует удалению из графа всех ребер, соответствующих x_i. Пусть алгоритм a порождается объектом x_i, тогда есть возможность, что после исключения x_i этот алгоритм перестанет быть достижимым из истоков графа. Оценим вклад a в оценку (2.2). Среди истоков графа обязательно найдется такой исток a_s, что $a_s \prec a$. Это означает, что будет выполнено $|A_{is}| = 0$. Если $I(a, x_i) \neq I(a_s, x_i)$, то $|B_{is}| \geq d(a_s, x_i) \geq d(x_i)$ в силу определения отступа. Если же $I(a, x_i) = I(a_s, x_i)$, то, поскольку x_i — порождающий объект для a, найдется алгоритм a_j, классификация которого отличается от a лишь на одном объекте, и для которого выполнено $|B_{js}| \geq d(x_i)$. Так как векторы ошибок a_i и a_j отличаются лишь в одном элементе, то $|B_{is}| \geq |B_{js}| - 1 \geq d(x_i) - 1$.

Поскольку в (2.2) берется минимум по всем истокам, вклад a в оценку не будет превосходить величины

$$\frac{C^{l-u}_{L-u-d(x_i)-1}}{C^l_L} \mathcal{H}^{l-u,\ m-d(x_i)-1}_{L-u-d(x_i)-1} \left(\frac{l}{L}(m - \varepsilon k) \right)$$

Эта величина быстро убывает при росте $d(x_i)$, поэтому, если $d(x_i)$ велико, то при исключении x_i мы потеряем только алгоритмы, не вносящие большого вклада в оценку.

Итак, если известны отступы $d(x)$ всех объектов, то, исключив объекты с большими отступами, можно значительно упростить перебор.

3.1 Вычисление отступов

Точное вычисление отступов является задачей не менее трудной, чем полный обход SC-графа, поэтому предлагается вычислять их приближенно.

Изначально все отступы полагаются равными n. Для каждого объекта $x \in \mathbb{X}$ определенное число раз случайным образом строится алгоритм, для которого x является порождающим. Пусть мы построили алгоритм a, тогда отступы всех объектов обновляются следующим образом:

$$d(x_i) := \min \left(d(x_i), \min \left\{ |B_{is}| \;\middle|\; a_s \in S, I(a, x_i) \neq I(a_s, x_i) \right\} \right) \tag{3.1}$$

Опишем подробнее, как указанный метод будет работать для семейств линейных классификаторов. Пусть мы зафиксировали объект x. Для него определенное число раз повторяется следующая процедура: случайным образом выбираются $d-1$ объектов $x_{i_1}, \ldots, x_{i_{d-1}}$, к ним добавляется x, и строится гиперплоскость, проходящая через

эти d объектов. Эта гиперплоскость соответствует нескольким алгоритмам, так как для нее существует два способа выбрать ориентацию и 2^d способов выбрать классификацию d порождающих объектов, причем для всех этих алгоритмов x будет порождающим объектов. Из этих алгоритмов случайным образом выбирается один (обозначим его a), и для всех объектов обновляются отступы по формуле (3.1).

3.2 Эксперименты

3.2.1 Зависимость t от зашумленности выборки

На рис. 1-11 приведены графики зависимости оценки Q_ε от t при разных степенях зашумленности выборки, а также функции распределения отступов для этих выборок. Зашумленность характеризуется величиной $s = \min_{a \in \mathcal{A}} m(a)$. Все эксперименты проводились на выборках с $L = 200$, $\ell = 100$, $d = 2$, $\varepsilon = 0.1$, каждый класс генерировался из нормального распределения. Пример выборки приведен на рис. 1.

Видно, что во всех случаях достаточно взять $t = s + 10$, чтобы либо получить хорошее приближение оценки, либо понять, что оценка слишком завышена и точное ее вычисление не имеет смысла.

Также на рис. 10 приведен аналогичный график для выборки с $L = 400$.

На рис. 2 изображены оценки (1) и (2), а также оценка полного скользящего контроля. Видно, что улучшенная оценка значительно менее завышена, нежели классическая.

3.2.2 Поведение оценки при увеличении размерности

Были проведены эксперименты на выборках в трехмерном пространстве с $L = 200$, $\ell = 100$, $\varepsilon = 0.1$. Результаты приведены на рис. 12-15.

Вычисления стали занимать существенно больше времени, более того, оценки стали сильно завышенными даже при небольших уровнях зашумленности.

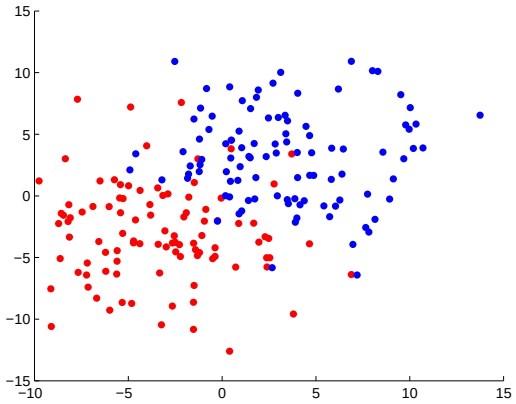


Рис. 1: Выборка с $s = 14$

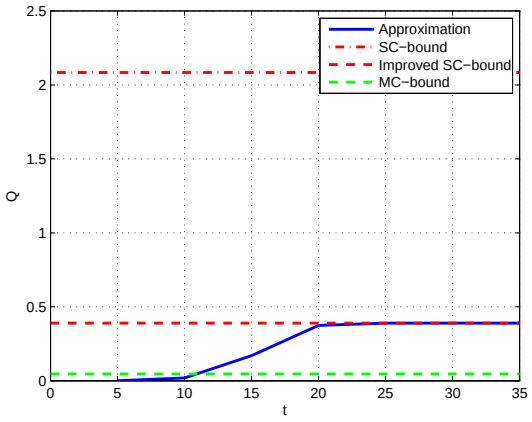


Рис. 2: Сравнение классической и улучшенной оценок расслоения-связности

3.3 Направления дальнейших исследований

- Применить описанную технику для ускорения случайного блуждания по графу; провести эксперименты, показывающие, что удаление части объектов слабо влияет на результат.

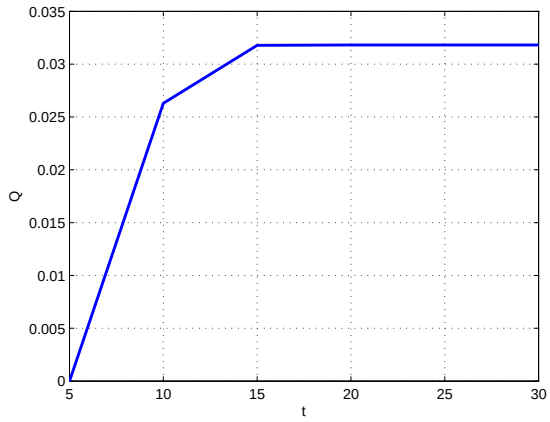


Рис. 3: $L = 200$, $d = 2$, $s = 0$

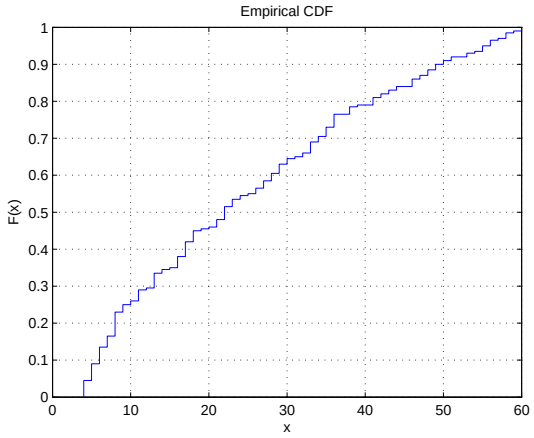


Рис. 4: $L = 200$, $d = 2$, $s = 0, CDF$

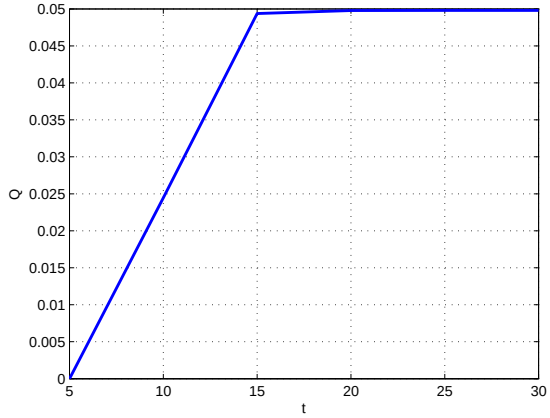


Рис. 5: $L = 200$, $d = 2$, $s = 1$

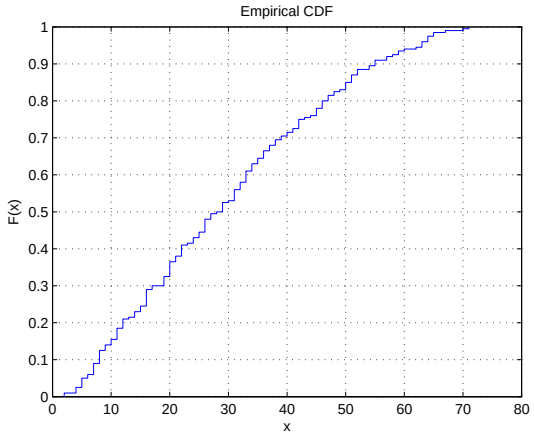


Рис. 6: $L = 200$, $d = 2$, $s = 1, CDF$

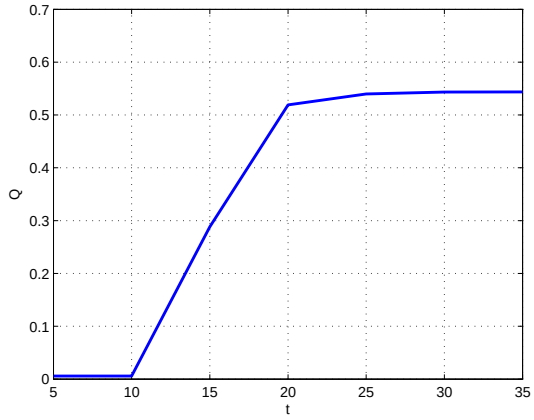


Рис. 7: $L = 200$, $d = 2$, $s = 8$

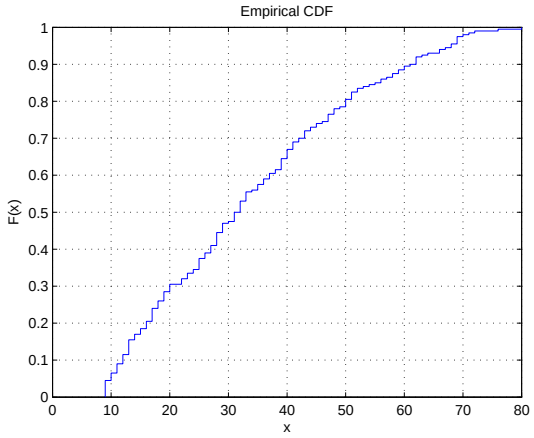


Рис. 8: $L = 200$, $d = 2$, $s = 8, CDF$

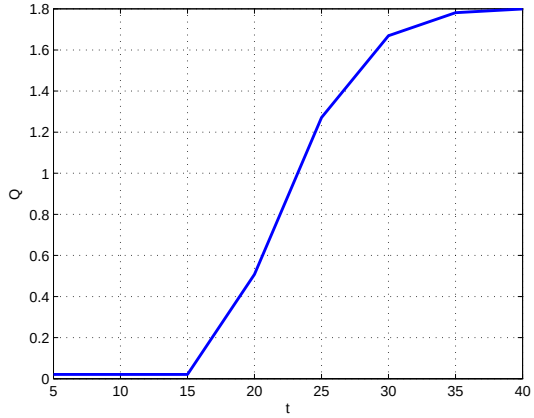


Рис. 9: $L = 200$, $d = 2$, $s = 14$

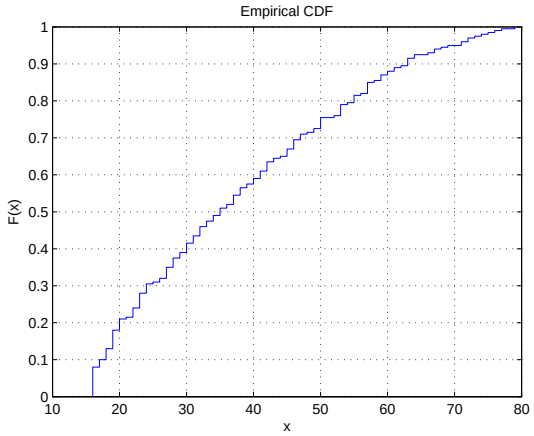


Рис. 10: $L = 200$, $d = 2$, $s = 14, CDF$

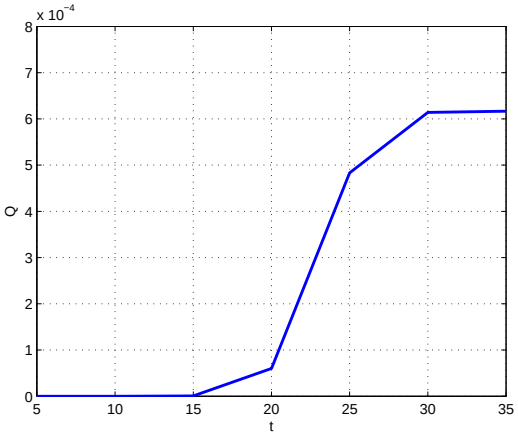


Рис. 11: $L = 400$, $d = 2$, $s = 6$

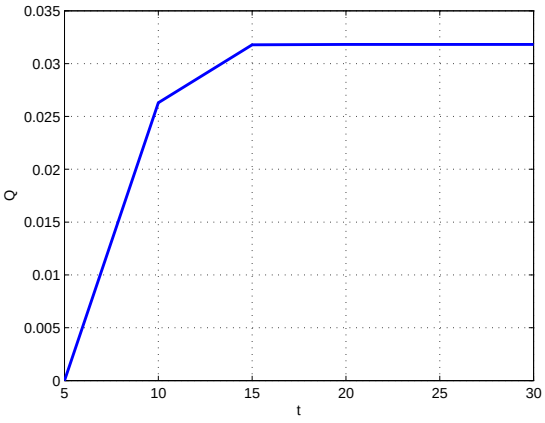


Рис. 12: $L = 200$, $d = 3$, $s = 0$

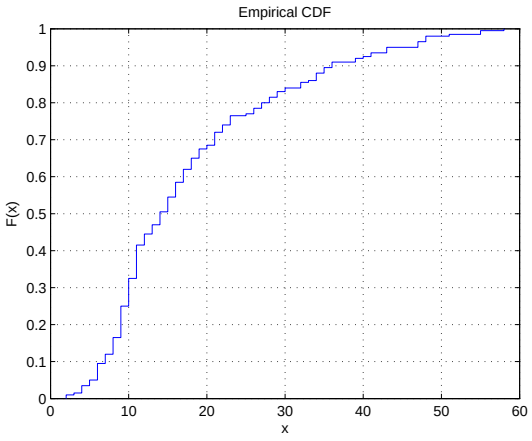


Рис. 13: $L = 200$, $d = 3$, $s = 0$, CDF

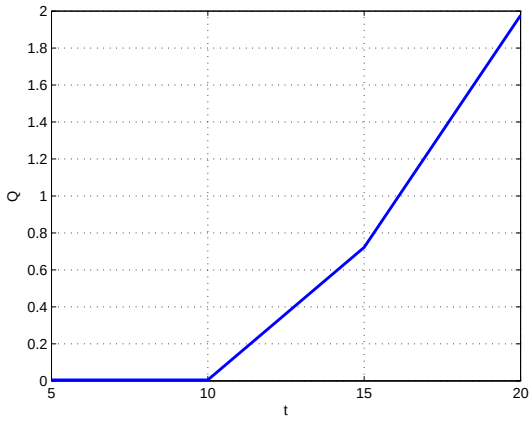


Рис. 14: $L = 200$, $d = 3$, $s = 7$

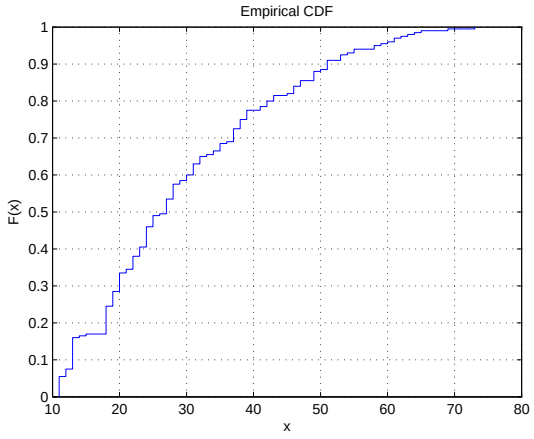


Рис. 15: $L = 200$, $d = 3$, $s = 7$, CDF

4 Приближенное вычисление оценки расслоения-связности

Обозначим вклад алгоритма $a \in \mathcal{A}$ в оценку (2.2) через $b(a)$.

Пусть имеется набор алгоритмов $a_1, \ldots, a_n$, выбранных равномерно из семейства $\mathcal{A}$. Тогда можно получить оценку для Q_ε:

$$\hat{Q}_\varepsilon = \frac{1}{n} \sum_{i=1}^{n} b(a_i)$$

Как говорилось выше, оценка Q_ε определяется в основном вкладами алгоритмов из нижних слоев графа расслоения-связности. Однако число алгоритмов в нижних слоях составляет лишь небольшую долю среди всех алгоритмов, поэтому вероятность того, что в выборке $a_1, \ldots, a_n$ встретится алгоритм из первых t слоев, крайне невысока. Чтобы сгенерировать выборку из первых t слоев, воспользуемся методом случайного блуждания по графу (см. алгоритм 4.1), применив его к нижним t слоям. Известно ([**?**]), что если граф не является двудольным, то вероятность получить алгоритм a на i-м шаге стремится при росте i к величине $\pi(a) = \frac{\deg(a)}{2|E|}$, где $|E|$ — число ребер в графе.

Алгоритм 4.1 Случайное блуждание

Вход: Граф $G = (V, E)$, стартовая вершина v_1, число итераций i_{max}

Выход: Выборка $v_1, v_2, \ldots, v_{i_{max}}$

1: **для** $i = 2, \ldots, i_{max}$

2: $\quad R := \{v \in V \mid (v_{i-1}, v) \in E\}$ // *окрестность вершины* v_{i-1}

3: $\quad$ Выбрать случайно вершину v' из равномерного распределения на R

4: $\quad v_i := v'$

Граф расслоения-связности является двудольным, поэтому для того, чтобы получить на нем то же стационарное распределение, необходимо применять *ленивое случайное блуждание* (см. алгоритм 4.2). Оно отличается от обычного случайного блуждания тем, что на каждом шаге с вероятностью $\frac{1}{2}$ блуждание остается на месте, в вершине с предыдущего шага. Это равносильно добавлению петель к каждой вершине, в результате чего граф перестает быть двудольным, и к нему применим указанный выше результат (вероятность получить вершину a стремится к $\pi(a)$).

Алгоритм 4.2 Ленивое случайное блуждание

Вход: Граф $G = (V, E)$, стартовая вершина v_1, число итераций i_{max}

Выход: Выборка $v_1, v_2, \ldots, v_{i_{max}}$

1: **для** $i = 2, \ldots, i_{max}$
2: Сгенерировать число r из распределения Бернулли с $p = \frac{1}{2}$
3: **если** $r = 0$ **то**
4: $R := \{v \in V \mid (v_{i-1}, v) \in E\}$ // *окрестность вершины* v_{i-1}
5: Выбрать случайно вершину v' из равномерного распределения на R
6: $v_i := v'$
7: **иначе**
8: $v_i := v_{i-1}$

Итак, с помощью случайного блуждания мы можем получить выборку $a_1, \ldots, a_n$ алгоритмов из первых t слоев, где, начиная с некоторого номера, алгоритм a появляется с вероятностью $\pi(a) = \frac{\deg(a)}{2|E|}$. Сделав поправку на эту вероятность, можно получить несмещенную оценку для среднего вклада по слою m:

$$\hat{Q}_\varepsilon = \frac{1}{n} \sum_{i=1}^{n} \frac{b(a_i)}{\pi(a_i)} \tag{4.1}$$

Вычислим матожидание данной оценки (здесь $\hat{a}_1, \ldots, \hat{a}_D$ — все алгоритмы семейства):

$$\begin{aligned}
\mathbb{E}\hat{Q}_\varepsilon &= \frac{1}{n} \sum_{i=1}^{n} \frac{b(a_i)}{\pi(a_i)} = \\
&= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\frac{b(a_i)}{\pi(a_i)} = \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{D} \frac{b(\hat{a}_j)}{\pi(\hat{a}_j)} \pi(\hat{a}_j) = \\
&= \frac{1}{n} \sum_{i=1}^{n} Q_\varepsilon = \\
&= Q_\varepsilon
\end{aligned}$$

Таким образом, оценка (4.1) действительно является несмещенной.

Отметим, что мы осуществляем блуждание только по первым t слоям графа, поэтому величина $|E|$ — это число ребер только в этих слоях, которое мы будем

обозначать через E_t. Для того, чтобы узнать точное значение E_t, необходимо полностью обойти первые t слоев, что крайне нежелательно. Предлагается вместо этого предлагается преобразовать величину $\pi(a)$:

$$\pi(a) = \frac{\deg(a)}{2E_t} = \frac{\deg(a)}{V_t \frac{E_t}{V_t}},$$

где V_t — число вершин в первых t слоях. Отношение $\frac{|E|}{|V|}$ является одинаковым практически для всех подмножеств графа расслоения-связности, поэтому можно подставить его вместо $\frac{E_t}{V_t}$. Величину V_t можно оценить путем случайной генерации алгоритмов из семейства $\mathcal{A}$.

Вклады в оценку $b(a)$ в пределах одного слоя сильно варьируются, и простое случайное блуждание может учесть эти вариации лишь при очень большом числе шагов. Существует метод Frontier Sampling ([**?**]), позволяющий избежать таких проблем (см. алгоритм 4.3). В данной работе в качестве стартовых вершин P предлагается брать множество всех истоков графа расслоения-связности. Показано, что если граф не является двудольным, то вероятность получить алгоритм a на i-м шаге стремится при росте i к величине

$$\pi(a) = \frac{deg(a)}{2|E|}$$

Если произвести модификацию алгоритма, аналогичную модификации в ленивом случайном блуждании, то можно получить такое же стационарное распределение и для двудольных графов.

Отметим, что в случайном блуждании следует отбрасывать некоторое количество первых сэмплов, поскольку в начале блуждания распределение марковской цепи отличается от $\pi(a)$.

С помощью алгоритма 4.3 была вычислена оценка расслоения-связности для линейно неразделимой выборки с $L = 200$. Наилучший алгоритм на данной выборке допускал 8 ошибок. Оценка вычислялась по формуле (4.1), причем в качестве E_t бралось истинное число ребер в первых t слоях. Результат изображен на рис. 16. Видно, что методу не удается получить несмещенную оценку. Это говорит о том, что сходимость метода крайней медленная и не достигается даже за 30000 тысяч итераций.

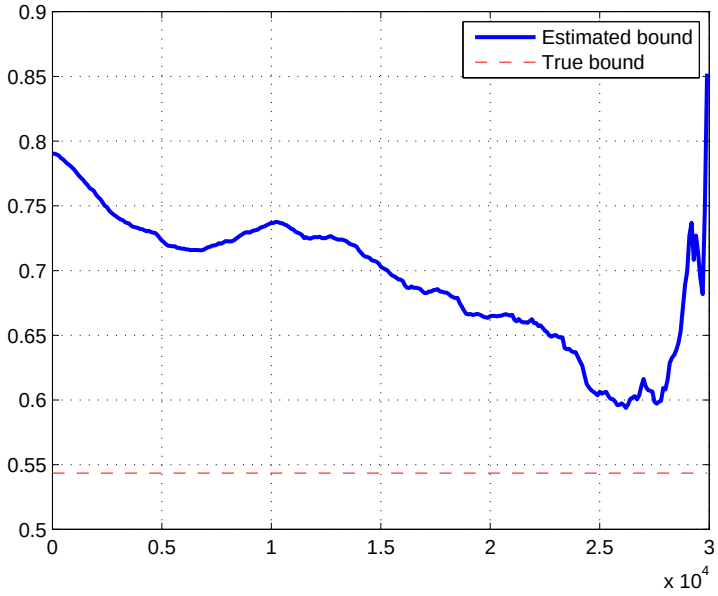


Рис. 16: Frontier sampling. По оси X — число отброшенных первых сэмплов, по оси Y — полученная при этом оценка.

Увеличить сходимость позволяет следующий прием. Преобразуем оценку вероятности переобучения:

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{i=1}^{D} b(a_i) = \sum_{m=0}^{L} |A_m| \left(\frac{1}{|A_m|} \sum_{a \in A_m} b(a) \right),$$

где A_m — это множество всех алгоритмов из $\mathcal{A}$, допускающих m ошибок. Тогда, если обозначить средний вклад в оценку алгоритмов из m-го слоя через $Q_m = \frac{1}{|A_m|} \sum_{a \in A_m} b(a)$, то получаем следующую запись для оценки:

$$Q_\varepsilon(\mu, \mathbb{X}) \leq \sum_{m=0}^{L} |A_m| Q_m \tag{4.2}$$

В качестве Q_m предлагается использовать следующую оценку:

$$\hat{Q}_m = \frac{1}{\sum_{i=1}^{n} [m(a_i) = m]} \sum_{i=1}^{n} [m(a_i) = m] \frac{1}{\pi(a_i)|V|} b(a_i) \tag{4.3}$$

Мощность m-го слоя $|A_m|$ предлагается оценивать следующим образом:

$$|\hat{A}_m| = \frac{V_t}{n} \sum_{i=1}^{n} [m(a_i) = m]$$

А лучше так:

$$|\hat{A}_m| = \frac{V_t}{n} \sum_{i=1}^{n} \frac{[m(a_i) = m]}{\pi(a_i)}$$

Алгоритм 4.3 Frontier sampling

Вход: Граф $G = (V, E)$, набор стартовых вершин $P = (v^1, \ldots, v^s)$, число итераций i_{max}

Выход: Выборка $v_1, v_2, \ldots, v_{i_{max}}$

1: **для** $i = 1, \ldots, i_{max}$

2: Выбрать вершину $v \in P$ с вероятностью $\frac{deg(v)}{\sum_{u \in P} deg(u)}$

3: $R := \{v' \in V \mid (v, v') \in E\}$ // *окрестность вершины* v

4: Выбрать случайно вершину v' из равномерного распределения на R

5: $v_i := v'$

6: Заменить в P вершину v на v'

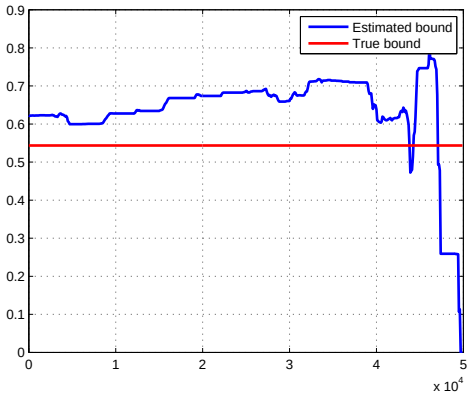


Рис. 17: Ленивое случайное блуждание. По оси X — число отброшенных первых сэмплов, по оси Y — полученная при этом оценка.

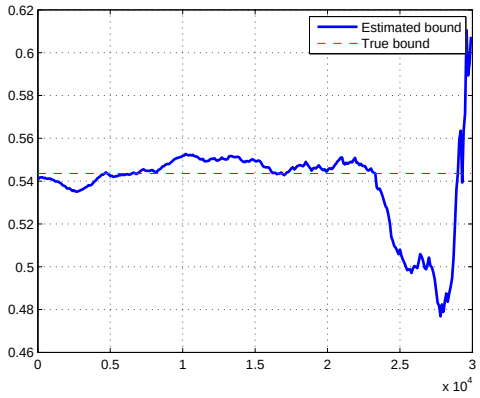


Рис. 18: Frontier sampling. По оси X — число отброшенных первых сэмплов, по оси Y — полученная при этом оценка.

Оценка, вычисленная данным методом для выборки, описанной выше, изображена на рис. 18. В качестве $|A_m|$ использовались истинные значения. В данном эксперименте удается получить несмещенные оценки (среднее по всем оценкам совпадает с истинной оценкой расслоения-связности). Также на рис. 17 изображена оценка, вычисленная с помощью обычного ленивого случайного блуждания. Даже с использованием описанного приема она не позволяет получить несмещенную оценку.

На рис. 19 показана оценка, вычисленная с помощью метода 4.3 с использованием приближения $|A_m|$. Видно, что возникает некоторое смещение оценки, но оно является достаточно небольшим (примерно 0.05) и все равно позволяет судить о величине вероятности переобучения.

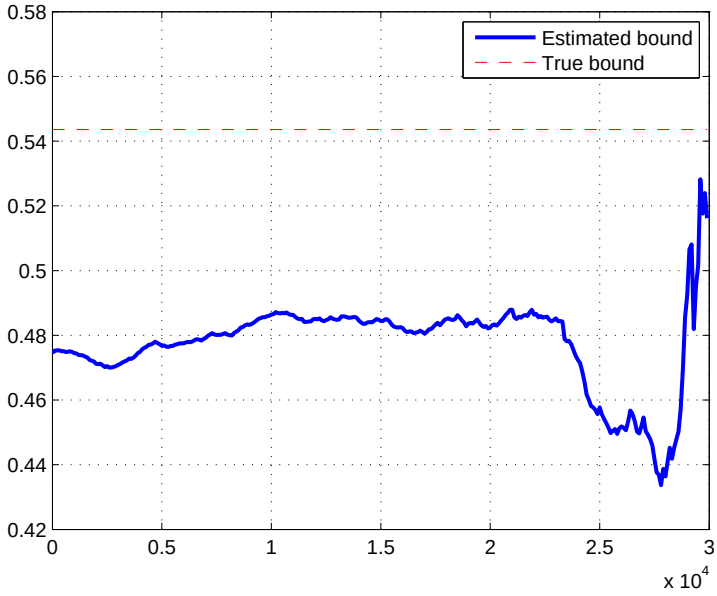


Рис. 19: Frontier sampling. По оси X — число отброшенных первых сэмплов, по оси Y — полученная при этом оценка. Использовалось приближение для $|A_m|$.

На рис. 20 изображены оценки, вычисленные по 1000 сэмплов, начиная с определенного номера. Видно, что оценки становятся несмещенными только для сэмплов с номером больше 10000, что говорит о небольшой скорости сходимости. Известно ([**?**]), что повышения сходимости можно добиться, если на каждом шаге случайного блуждания с небольшой вероятностью α переходить в равномерно выбранную вершину графа. Поскольку в нашем случае рассматриваются нижние слои графа расслоения-связности, то равномерной генерации можно добиться путем обучения по случайной подвыборке.

4.1 Отбор признаков

Эксперименты проводились на наборе данных Ecoli из репозитория UCI. В данных имелось два категориальных признака, которые были исключены. Проекции выборки на двухэлементные подмножества признаков изображены на рис. 21.

Для всех двухэлементных подмножеств признаков были вычислены два критерия качества: основанный на комбинаторных оценках и основанный на регуляризации.

Критерий, основанный на регуляризации, вычислялся по формуле

$$Q_r = \nu(a, X) + \|w\|^2,$$

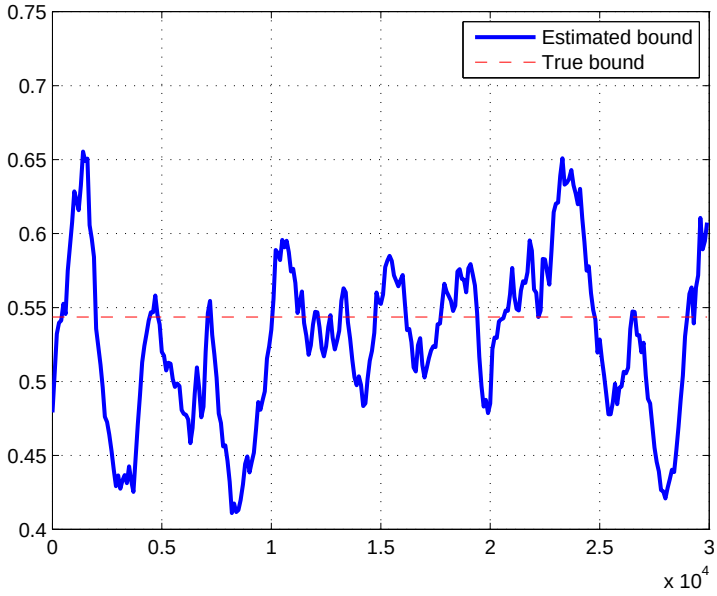


Рис. 20: Frontier sampling. По оси X — номер сэмпла, начиная с которого была взята тысяча сэмплов, по оси Y — полученная при этом оценка.

где a — линейный алгоритм, построенный методом SVM, w — соответствующий ему вектор весов.

Вычисление комбинаторного критерия состояло из следующих шагов:

1. Строилась линейная разделяющая поверхность методом SVM.
2. Из нее осуществлялся спуск вниз по SC-графу до истока.
3. Из этого истока запускался обход всех слоев SC-графа вплоть до $(m_0 + 3)$-го, где m_0 — число ошибок найденного истока; затем фиксировались все истоки, найденные во время этого обхода.
4. Оценивался профиль расслоения путем случайной генерации 20000 объектов.
5. Генерировалась 1000 объектов из 20 нижних слоев графа с помощью случайного блуждания.
6. Производилось обращение оценки с $\eta = \frac{1}{2}$.
7. вычислялся комбинаторный критерий по формуле

$$Q_c = \nu(a_0, X) + \varepsilon\left(\frac{1}{2}\right),$$

где a_0 — лучший алгоритм в семействе.

Величины, полученные с помощью описанных критериев, приведены в таблицах 1 и 2. Подмножества, отсортированные по качеству с точки зрения этих критериев, записаны в таблице 3.

	1	2	3	4	5
1		0.3214	0.306	0.142	0.217
2			0.327	0.128	0.235
3				0.229	0.339
4					0.119
4					

Таблица 1: Качество двухэлементных подмножеств с точки зрения комбинаторного критерия

	1	2	3	4	5
1		112.5	113	88.9	107.9
2			155.7	93	131.6
3				107.5	129.8
4					109.1
4					

Таблица 2: Качество двухэлементных подмножеств с точки зрения регуляризационного критерия

4.2 Направления дальнейших исследований

- Исключить из описанного подхода вычисление средних оценок по слоям, перейти к непосредственной оценке вероятности переобучения путем случайного блуждания;
- Ускорить сходимость, добавив небольшое количество обучений по случайным подвыборкам;
- Продолжить эксперименты с отбором признаков;

	Комбинаторный критерий	Регуляризационный критерий
1	$\{4, 5\}$	$\{1, 4\}$
2	$\{2, 4\}$	$\{2, 4\}$
3	$\{1, 4\}$	$\{3, 4\}$
4	$\{1, 5\}$	$\{1, 5\}$
5	$\{3, 4\}$	$\{4, 5\}$
6	$\{2, 5\}$	$\{1, 2\}$
7	$\{1, 3\}$	$\{1, 3\}$
8	$\{1, 2\}$	$\{3, 5\}$
9	$\{2, 3\}$	$\{2, 5\}$
10	$\{3, 5\}$	$\{2, 3\}$

Таблица 3: Подмножества, отсортированные по убыванию качества

- Встроить в эксперименты построений композиций.

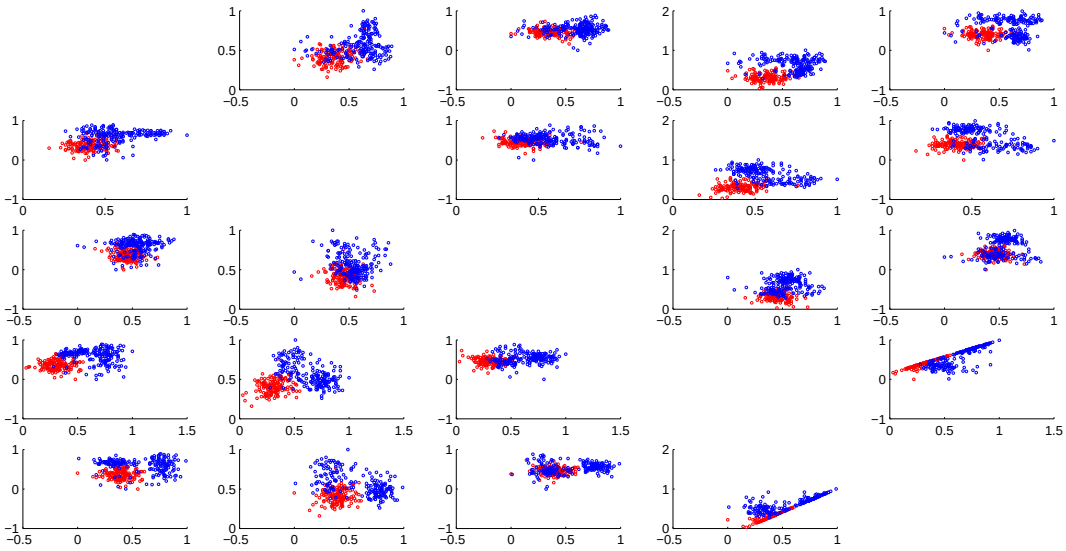


Рис. 21: Двухэлементные подмножества признаков

5 Общий метод обхода графа расслоения-связности

Пусть заданы выборка $\mathbb{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ и целевая функция $y : \mathbb{X} \to \mathbb{Y}$. Будем считать, что семейство алгоритмов задается следующим образом:

$$\mathcal{A} = \{a_\theta(x) \,|\, a_\theta(x) = [K(x, \theta) < 0], \theta \in \Theta\},$$

где $\Theta = \mathbb{R}^p$ — множество параметров, $K(x, \theta)$ — некоторая фиксированная функция.

Для каждого объекта $x_i \in \mathbb{X}$ определим гиперповерхность S_i в Θ:

$$S_i = \{\theta \in \Theta \,|\, K(x_i, w) = 0\}$$

Набор гиперповерхностей $\{S_1, \ldots, S_n\}$ задает разбиение Θ на максимальные связные области, не пересекающиеся ни с одной из этих гиперповерхностей. Такие области называются *ячейками*, а набор ячеек — *конфигурацией* гиперповерхностей. Основное свойство такого разбиения состоит в том, что все алгоритмы, лежащие в одной ячейке, одинаково классифицируют всю выборку. Значит, найдя все ячейки конфигурации, мы получим описание всего семейства $\mathcal{A}$.

5.1 Известные результаты

Пусть любая гиперповерхность S_i представима в виде

$$S_i = (Q_i = 0) \wedge F_i\left(P_{i_1}\sigma_{i_1}0, \ldots, P_{i_u}\sigma_{i_u}0\right),$$

где F_i — булева формула, $Q_i, P_{i_1}, \ldots, P_{i_u}$ — полиномы над Θ, $\sigma_{i_j} \in \{\leq, \geq\}$.

Известно, что в этом случае можно найти *разбиение на цилиндрические ячейки*, которое является подразбиением ячеек конфигурации [?].

Также существует алгоритм поиска ячеек для более общего случая, когда все S_i являются полупфаффовыми множествами [?]. Примерами полупфаффовых множеств являются графики тригонометрических, экспоненциальных и логарифмических функций [?].

5.2 Обход нижних слоев графа расслоения-связности

Мы сосредоточимся на простых семействах алгоритмов и получим для них алгоритм обхода нижних слоев SC-графа.

Предъявим следующие *требования регулярности* к $\mathcal{A}$:

1. Одному классу эквивалентности алгоритмов соответствует ровно одна ячейка.
2. Для любой размерности d существует такое число $m = m(d)$, что по любым m объектам выборки можно построить единственный алгоритм $a_\theta(x)$, из которого малыми изменениями θ можно получить любую классификацию этих m точек, не изменив классификацию остальных точек.
3. Для любого класса эквивалентности a найдется m объектов, по которым можно построить алгоритм, лежащий в этом классе. Множество таких наборов будем обозначать через T_a.
4. T_a является 1-связным, то есть из любого его набора можно получить любой другой, меняя только по одному элементу на каждом шаге, и не выходя при этом за пределы T_a.
5. Если a и a' — соседние алгоритмы, то $T_a \cap T_{a'} \neq \emptyset$.

Примерами семейств алгоритмов, удовлетворяющих этим требованиям, являются:

- конъюнкции: $K(x, \theta) = \prod_{i=1}^{d} [x_i < \theta_i]$
- линейные классификаторы: $K(x, \theta) = \langle x, \theta \rangle$
- шары: $K(x, \theta) = K(x, \theta_1, \theta_2) = \rho(x, \theta_1) - \theta_2$
- SVM: $K(x, \theta) = \sum_{i=1}^{h} \theta_i y_i M(x, x_i) - \theta_0$

Итак, пусть $\mathcal{A}$ — семейство алгоритмов, удовлетворяющее указанным выше требованиям, и пусть известен лучший или почти лучший алгоритм a_0. Тогда, используя алгоритм 5.1 для поиска соседних вершин, можно осуществить обход графа расслоения-связности, начиная с a_0.

5.3 Направления дальнейших исследований

- Показать для указанных семейств, что они действительно удовлетворяют условиям регулярности;

Алгоритм 5.1 Построение окрестности алгоритма a

Вход: $\mathbb{X}$, a;

Выход: V_a — окрестность алгоритма a;

1: Найти набор $D_0 = (x_{i_1}, \ldots, x_{i_m})$, по которому можно построить алгоритм, эквивалентный a

2: $T_a := \{D_0\}$

3: D_0.проверен := нет

4: $V_a := \emptyset$

5: **пока** $\exists D \in T_a :\ D$.проверен = нет

6: **для всех** $x_j \in D$

7: **для всех** $t \in \{1, \ldots n\} \setminus D$

8: $D' := (D \setminus \{x_j\}) \cup x_t$

9: $\hat{\theta} := (D')$

10: **если** из $\hat{\theta}$ можно получить алгоритм с таким же вектором ошибок, как у a **то**

11: $T_{a_0} := T_{a_0} \cup \{D'\}$

12: D'.проверен := нет

13: **если** из $\hat{\theta}$ можно получить алгоритм, вектор ошибок которого отличается от a ровно на одном объекте **то**

14: $V_a := V_a \cup \{D'\}$

15: D.проверен := да

- Попытаться увеличить эффективность алгоритма.